

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for debugging ~~in an application program~~
programs, the method comprising:

allocating prescribed regions of a system memory for each of a plurality of
application programs;

writing application identifier information on ~~a task~~ an application program
to be performed;

checking whether the ~~task~~ application program is performed in a designated
region; and

generating an interrupt signal when the ~~task~~ application program is
performed in a region other than the designated region for the ~~task~~ application program.

2. Canceled

3. (Currently Amended) The method according to claim 2, further comprising:

latching a data signal corresponding to the written application identifier information on the ~~task~~ application program; and
outputting a ~~task~~ an application signal corresponding to the ~~task~~ application identifier that is identified based on the latched data signal.

4. (Currently Amended) The method according to claim 3, wherein the designated region is an operation region assigned to each ~~task~~ application program.

5. (Currently Amended) The method according to claim 1, further comprising:

determining the information on the ~~task~~ application responsive to the interrupt signal; and
performing an operation corresponding to the information on the ~~task~~ application based on the determination.

6. (Currently Amended) The method according to claim 5, further comprising sending a control signal that is generated based on the interrupt signal to ~~a~~ the system memory.

7. (Currently Amended) The method according to claim 1, wherein after a ~~task~~ an application program switching occurs to a next ~~task~~ application, a checking process based on information corresponding to the next ~~task~~ application is repeatedly conducted to check whether the next ~~task~~ application is performed in a corresponding designated region.

8. (Currently Amended) The method according to claim 1, wherein the checking whether the ~~task~~ application program is performed in a designated region further comprises:

checking an operation region of a current ~~task~~ application program being accessed and outputting a result;

generating an address signal corresponding to the checking result; and

outputting a grant signal based on the address signal.

9. (Original) The method according to claim 8, wherein a task signal is used for outputting the grant signal.

10-14. Canceled

15. (Currently Amended) A apparatus for debugging ~~in an application program~~ programs, the apparatus comprising:

first control means for writing a ~~task~~ an application identifier provided for ~~each task~~ a plurality of application programs to be performed, for generating a data signal corresponding to the ~~task~~ application identifier, and for activating a ~~selected task~~ application programs;

~~task~~ checking means for outputting a ~~task~~ an application signal corresponding to the ~~task~~ application identifier that is identified based on the data signal, and for generating an interrupt signal according to a determination whether a current ~~task~~ application program is performed in a designated region; and

storage means for writing the ~~task~~ application identifier provided by the first control means, and for assigning ~~an~~ a corresponding designated operation region ~~to each task~~ for each of the plurality of applications in said storage means; and

second control means for outputting a control signal to control the storage means based on the generated interrupt signal.

16. Canceled

17. (Currently Amended) The apparatus according to claim 15, wherein an address signal is used as a basis of determining whether the current ~~task~~application is performed in the designated region.

18. (Currently Amended) The apparatus according to claim 15 wherein the task checking means comprises:

a latching unit that latches the data signal;

a decoding unit that identifies the ~~task~~application identifier based on the latched data signal, and generates the ~~task~~application signal corresponding to the ~~task~~application identifier; and

a ~~task~~comparing ~~unit~~device that receives the ~~task~~application signal and generates the interrupt signal according to the address signal.

19. (Currently Amended) The apparatus according to claim 18, wherein a plurality of task comparing units is included in the comparing device that are equal in number to ~~tasks~~ application programs to be performed, wherein each comparing unit comprises,

a first logic gate that logically processes the address signal; and

a second logic gate that logically processes the task signal, an output signal of the first logic gate and a first write enable signal to output a grant signal used to generate the interrupt signal.

20. (New) The apparatus of claim 19, wherein the first logic gate is an OR-NOT gate and the second logic gate is an AND gate.

21. (New) The apparatus according to claim 15, wherein the control signal terminates a corresponding application program.

22. (New) The method of claim 9, wherein the outputting a grant signal comprises:

logically processing the address signal, an application signal and a first write enable signal in each of a plurality of application comparison units equal in number to the plurality of application programs, and wherein said logically processing comprises,

logically processing the address signal in a first logic gate, and

logically processing the application signal, an output signal of the first logic gate and a first write enable signal in a second logic gate to output the grant signal used to generate the interrupt signal.